Lab 1. Image Reading and Writing operations using 8-bit Gray Scale BMP images

AIM: To perform Digital Image Negative for a given image

EXPLANATION:
The negative of an image with gray levels in the range [0, L-1] is obtained by using negative transformation which is given by the expression
$s = L - 1 - r$
Where s is output negative value, L is total gray levels and r is input pixel value

Details of 8-bit BMP file header:
14 bytes file header
40 bytes info header
1024 bytes color format
= 1078 bytes total header
and then data starts (.i.e first pixel....till last pixel of the image)

PROCEDURE
1. START
2. Open source file (say, in_vsat.bmp) and destination file (say out_vsat.bmp).
3. Read the header (1078 bytes) of the source file (in_vsat.bmp)
4. Write the header (1018 bytes) to destination file (out_vsat.bmp).
5. Read each image pixel (say, f (x)) of source file till end of the file.
6. Computer $s = L - 1 - f(x)$
7. Write s to destination file.
8. Close the source file (say, in_vsat.bmp) and destination file (say out_vsat.bmp).
9. Display the destination file
10. END

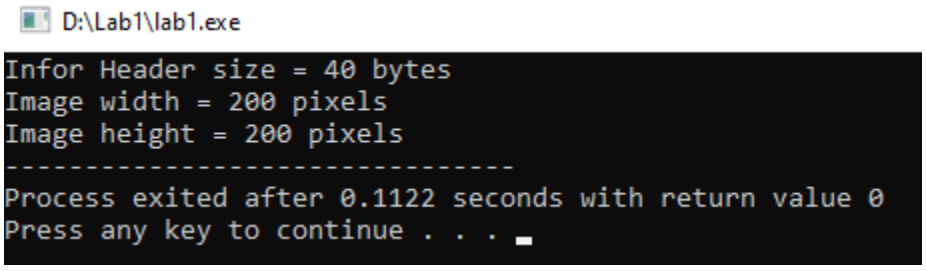
CONCLUSION
Digital Negative of a given image was created. It is useful for finding faults.

Program Code:

```
// Image Negative using fread and fwrite in C++
#include<stdio.h>
#include<conio.h>
main()
{
   FILE *sfile;
   FILE *dfile;

   char pixel;
   long size, height, width;

   sfile = fopen("giraffe.bmp", "rb");
   dfile = fopen("giraffeNegative.bmp", "wb");

   //Reading Heading to read width and height of the input image.
   for(int i = 0; i < 1078; i++)
   {
      if(i == 14 && i <= 25)
      {
            fread(&size,sizeof(long),1,sfile);
            fwrite(&size,sizeof(long),1,dfile);

            fread(&width,sizeof(long),1,sfile);
            fwrite(&width,sizeof(long),1,dfile);

              fread(&height,sizeof(long),1,sfile);
            fwrite(&height,sizeof(long),1,dfile);
      }
      else
      {
            fread(&pixel,sizeof(char),1,sfile);
            fwrite(&pixel,sizeof(char),1,dfile);
      }
```

```
    }

    //Making Negative and writing to destination file.
    while(fread(&pixel, sizeof(char), 1, sfile) != NULL)
    {
        pixel = 255 - pixel;
        fwrite(&pixel, sizeof(char), 1, dfile);
    }

fclose(sfile);
fclose(dfile);

            printf("Infor Header size = %d bytes\n", size);
    printf("Image width = %d pixels\n", width);
    printf("Image height = %d pixels", height);
    // getch();

} // End of File
```

Program Output:



giraffe.bmp

giraffeNegative.bmp